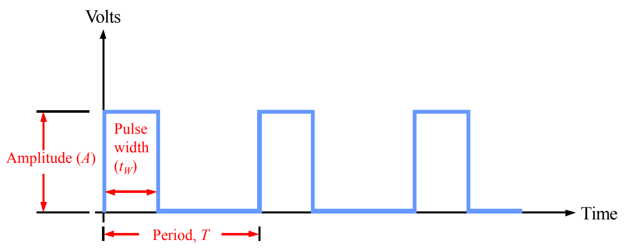
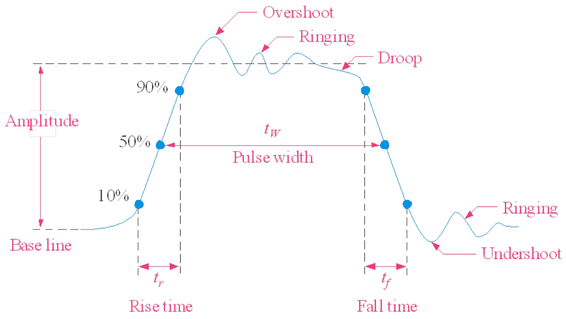
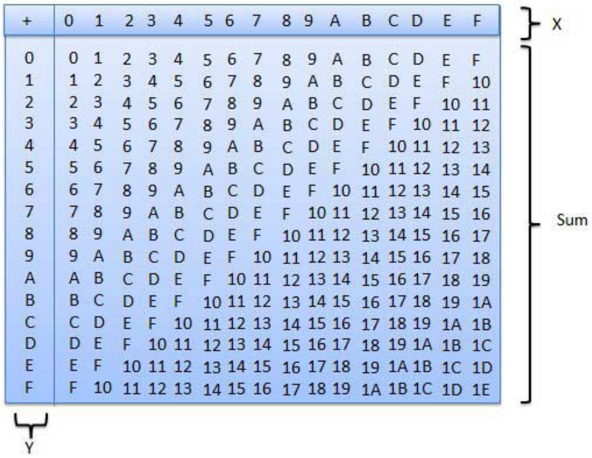




$$T = \frac{1}{f} \qquad f = \frac{1}{T}$$

Hex	Binary	Decimal	BCD
0	0000	0	0000
1	1	1	0001
2	10	2	0010
3	11	3	0011
4	100	4	0100
5	101	5	0101
6	110	6	0110
7	111	7	0111
8	1000	8	1000
9	1001	9	1001
A	1010	10	0001 0000
B	1011	11	0001 0001
C	1100	12	0001 0010
D	1101	13	0001 0011
E	1110	14	0001 0100
F	1111	15	0001 0101



+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
512	256	128	64	32	16	8	4	2	1	0.5	0.25	0.125	0.0625	.03125	.015625
8^9	8^8	8^7	8^6	8^5	8^4	8^3	8^2	8^1	8^0	8^{-1}	8^{-2}	8^{-3}	8^{-4}	8^{-5}	8^{-6}
134,217,728	16,777,216	2,097,152	262,144	32,768	4096	512	16	8	1	0.125	0.015625				
16^9	16^8	16^7	16^6	16^5	16^4	16^3	16^2	16^1	16^0	16^{-1}	16^{-2}	16^{-3}	16^{-4}	16^{-5}	16^{-6}
		268,435,456	16777216	1048576	65536	4096	256	16	1						

Two's Complement Addition

Add the values and discard any carry-out bit.

Examples: using 8-bit two's complement numbers.

```
Add −8 to +3
 (+3)   0000 0011
+(−8)   1111 1000
-----------------
 (−5)   1111 1011
Add −5 to −2
 (−2)   1111 1110
+(−5)   1111 1011
-----------------
 (−7) 1 1111 1001 : discard carry-out
```

Overflow Rule for addition

If 2 Two's Complement numbers are added, and they both have the same sign (both positive or both negative), then overflow occurs if and only if the result has the opposite sign. Overflow never occurs when adding operands with different signs.

i.e. Adding two positive numbers must give a positive result
Adding two negative numbers must give a negative result

Overflow occurs if

(+A) + (+B) = −C
(−A) + (−B) = +C

Example: Using 4-bit Two's Complement numbers (−8 ≤ x ≤ +7)

```
 (−7)   1001
+(−6)   1010
------------
(−13) 1 0011 = 3 : Overflow (largest −ve number is −8)
```

A couple of definitions:

Subtrahend: what is being subtracted
Minuhend: what it is being subtracted from

Example: 612 - 485 = 127

485 is the subtrahend, 612 is the minuhend, 127 is the result

Two's Complement Subtraction

Normally accomplished by negating the subtrahend and adding it to the minuhend. Any carry-out is discarded.

Example: Using 8-bit Two's Complement Numbers (−128 ≤ x ≤ +127)

```
 (+8) 0000 1000              0000 1000
−(+5) 0000 0101 -> Negate -> +1111 1011
-----                        -----------
 (+3)                      1 0000 0011 : discard carry-out
```

Overflow Rule for Subtraction

If 2 Two's Complement numbers are subtracted, and their signs are different, then overflow occurs if and only if the result has the same sign as the subtrahend.

Overflow occurs if

(+A) − (−B) = −C
(−A) − (+B) = +C

Example: Using 4-bit Two's Complement numbers (−8 ≤ x ≤ +7)

Subtract −6 from +7

```
  (+7) 0111            0111
 −(−6) 1010 -> Negate -> +0110
 ----------            -----
   13                  1101 = −8 + 5 = −3 : Overflow
```

Binary Addition

It is a key for binary subtraction, multiplication, division. There are four rules of binary addition.

Case	A	+	B	Sum	Carry
1	0	+	0	0	0
2	0	+	1	1	0
3	1	+	0	1	0
4	1	+	1	0	1

0011010 + 001100 = 00100110

$$\begin{array}{rl} 1\,1 & \text{carry} \\ 0011010 & = 26_{10} \\ +0001100 & = 12_{10} \\ \hline 0100110 & = 38_{10} \end{array}$$

Binary Subtraction

Subtraction and Borrow, these two words will be used very frequently for the binary subtraction. There are four rules of binary subtraction.

Case	A	-	B	Subtract	Borrow
1	0	-	0	0	0
2	1	-	0	1	0
3	1	-	1	0	0
4	0	-	1	0	1

0011010 - 001100 = 00001110

$$\begin{array}{rl} 1\,1 & \text{borrow} \\ 00\cancel{11}010 & = 26_{10} \\ -0001100 & = 12_{10} \\ \hline 0001110 & = 14_{10} \end{array}$$

Binary Multiplication

Binary multiplication is similar to decimal multiplication. It is simpler than decimal multiplication because only 0s and 1s are involved. There are four rules of binary multiplication.

Case	A	x	B	Multiplication
1	0	x	0	0
2	0	x	1	0
3	1	x	0	0
4	1	x	1	1

Example:

0011010 x 001100 = 100111000

$$\begin{array}{rl} 0011010 & = 26_{10} \\ \times 0001100 & = 12_{10} \\ \hline 0000000 & \\ 0000000 & \\ 0011010 & \\ 0011010 & \\ \hline 0100111000 & = 312_{10} \end{array}$$

Binary Division

Binary division is similar to decimal division. It is called as the long division procedure.

$$\begin{array}{rl} 111 & = 7_{10} \\ 000110\,\overline{)\,\cancel{1}\overset{1}{0}1010} & = 42_{10} \\ -110 & = 6_{10} \\ \hline \cancel{1}\overset{1}{0}01 & \\ -110 & \\ \hline 110 & \\ -110 & \\ \hline 0 & \end{array}$$

Octal Addition

$456_8 + 123_8 = 601_8$

$$\begin{array}{rl} 1\,1 & \text{carry} \\ 456 & = 302_{10} \\ +123 & = 83_{10} \\ \hline 601 & = 385_{10} \end{array}$$

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

A

Sum

B

Two's Complement to Binary

Here is the process to convert a negative two's complement number back to decimal:

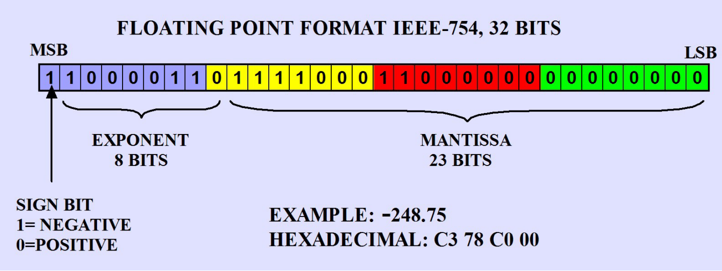
(1) flip all the bits,
(2) add 1, and
(3) interpret the result as a binary representation of the magnitude and add a negative sig

So, for example, we have:

1111 1111 1011 0101
→ 0000 0000 0100 1010
→ 0000 0000 0100 1011
→ −75

Decimal to Floating Point Conversion

1. Sign bit, 0 for positive, 1 for negative.
2. Convert whole number part to binary.
3. Multiply part past decimal point by 2 until no more remains past the decimal.
4. Collect numbers from top to bottom.
5. Move radix until 1 away from the front.
6. However many spaces you moved it, that's what the exponent is equal to.
7. Add 127 to the exponent.
8. Add 0's to fill up the rest of the mantissa.



Binary to Hexadecimal

Large binary number can easily be converted to hexadecimal by grouping bits 4 at a time and writing the equivalent hexadecimal character.

Express 1001 0110 0000 11102 in hexadecimal:
Group the binary number by 4-bits starting from the right. Thus, 960E

Hexadecimal to Binary

Express 1A2F16 in decimal
Start by writing the column weights:

4096 256 16 1
1 A 2 F16
1(4096) + 10(256) +2(16) +15(1) = 6703_{10}

Decimal to Hexadecimal

The hexadecimal number 2C has a decimal equivalent value of 44